

CLAIMS

1 1. A method for facilitating profiling of an application, comprising:
 2 intercepting application instructions immediately before their execution;
 3 determining if an application instruction is a frequently used instruction
 4 according to a pre-established policy; and
 5 only if the application instruction is a frequently used instruction,
 6 instrumenting the application instruction so as to facilitate collection of information
 7 about execution of the application instruction.

1 2. The method of claim 1, wherein the intercepted application instructions
 2 are application binaries.

1 3. The method of claim 1, wherein determining if an application
 2 instruction is a frequently used instruction comprises consulting a program counter
 3 associated with the application instruction.

1 4. The method of claim 1, wherein instrumenting the application
 2 instruction comprises instrumenting the application instruction to collect information
 3 as to the fact that the application instruction was executed.

1 5. The method of claim 1, wherein instrumenting the application
 2 instruction comprises instrumenting the application instruction to collect information
 3 as to what other application instruction called the intercepted application instruction.

1 6. The method of claim 1, wherein instrumenting the application
2 instruction comprises instrumenting the application instruction to collect information
3 as to what application instructions the intercepted application instruction calls.

1 7. The method of claim 1, wherein instrumenting the application
2 instruction comprises instrumenting the application instruction to increment a counter
3 representing a number of processor cycles or a counter representing the number of
4 instructions executed.

1 8. The method of claim 1, further comprising recording information as to
2 the execution of the intercepted application instructions.

1 9. The method of claim 8, wherein recording information comprises
2 recording information as to the execution of code stored in a shared library that the
3 application accesses.

1 10. The method of claim 1, further comprising determining if code
2 associated with the application instructions has been cached.

1 11. The method of claim 10, further comprising executing the cached code
2 in lieu of the intercepted application instructions if associated code has been cached.

1 12. A system for facilitating profiling of an application, comprising:
 2 means for intercepting application instructions before they are executed;
 3 means for determining if an application instructions are used frequently; and
 4 means for instrumenting frequently used application instructions to facilitate
 5 collection of information about execution of the application instructions.

1 13. The system of claim 12, wherein the means for determining if an
 2 application instruction is a frequently used instruction comprise means for counting
 3 the number of times the application instruction is executed.

1 14. The system of claim 12, wherein the means for instrumenting the
 2 application instruction comprise means for instrumenting the application instruction
 3 to collect information as to at least one of the fact that the application instruction was
 4 executed, what other application instruction called the intercepted application
 5 instruction, and other application instructions the intercepted application instruction
 6 calls.

1 15. The system of claim 12, wherein the means for instrumenting the
 2 application instruction comprise means for instrumenting the application instruction
 3 to increment a counter representing a number of processor cycles.

1 16. The system of claim 12, further comprising means for recording
 2 information as to the execution of the intercepted application instructions.

1 17. A program that facilitates profiling of an application and that is stored
2 on a computer-readable medium, the program comprising:
3 logic configured to intercept application binaries;
4 logic configured to determine if an application instruction is a frequently used
5 instruction; and
6 logic configured to instrument the application instructions that are determined
7 to be frequently used instructions so as to facilitate collection of information about
8 execution of the application instruction.

1 18. The program of claim 17, wherein the logic configured to determine if
2 an application instruction is a frequently used instruction comprises logic configured
3 to count the number of times the application instruction is executed.

1 19. The program of claim 17, wherein the logic configured to instrument
2 the application instruction comprises logic configured to instrument the application
3 instruction to collect information as to at least one of the fact that the application
4 instruction was executed, what other application instruction called the intercepted
5 application instruction, and other application instructions the intercepted application
6 instruction calls.

1 20. The program of claim 17, wherein the logic configured to instrument
2 the application instruction comprises logic configured to instrument the application
3 instruction to increment a counter representing a number of processor cycles.

1 21. The program of claim 17, further comprising logic configured to
2 record information as to the execution of the intercepted application instructions.

1 22. A method for facilitating profiling of an application, comprising:
2 intercepting application binaries prior to their execution;
3 determining if associated code has been cached;
4 executing the cached code if associated code has been cached;
5 recording information about the execution of the application binaries if
6 associated code has not been cached;
7 instrumenting the application binaries if they are determined to be frequently
8 executed binaries; and
9 caching code associated with the application binaries that includes
10 instrumentation.

1 23. The method of claim 22, wherein instrumenting the application
2 binaries comprises instrumenting the application binaries to collect information as to
3 the fact that the application binaries were executed.

1 24. The method of claim 22, wherein instrumenting the application
2 binaries comprises instrumenting the application binaries to collect information as to
3 what other application binaries called the intercepted application binaries.

1 25. The method of claim 22, wherein instrumenting the application
2 binaries comprises instrumenting the application binaries to collect information as to
3 other application binaries the intercepted application binaries call.

1 26. The method of claim 22, wherein instrumenting the application
2 binaries comprises instrumenting the application binaries to increment a counter
3 representing a number of processor cycles.

1 27. The method of claim 22, wherein recording information comprises
2 recording information as to the execution of code stored in a shared library that the
3 application accesses.

1 28. A method for facilitating profiling of an application, comprising:
2 intercepting application code fragments prior to their execution;
3 determining if code associated with the fragments has been cached;
4 executing the cached code if associated code has been cached;
5 determining the number of very long instruction words (VLIWs) being
6 executed for each code fragment;
7 instrumenting the application code fragments if they are determined to be
8 frequently executed fragments; and
9 caching code associated with the code fragments that includes instrumentation.

1 29. The method of claim 28, wherein instrumenting the code fragments
2 comprises instrumenting the fragments to collect information as to the fact that the
3 fragments were executed.

1 30. The method of claim 28, wherein instrumenting the code fragments
2 comprises instrumenting the fragments to increment a counter representing a number
3 of processor cycles.